TP 3 Data Mining: KDE Kernel Density Estimation

Tuesday 24th March, 2020
deadline: Monday 13th April, 2020, 23:59
OBLIGATORY
Individual work

Kernels

The basic idea in Probability Density Estimation of an unknown density function $p(\mathbf{x})$ is to estimate the probability P of a given region R from a limited number, n, of training examples $\{\mathbf{x}_i | i = 1..n\}$. In theory this probability is given by:

$$P(\mathbf{x} \in R) = \int_R p(\mathbf{x}) d\mathbf{x}$$

For a large number of samples n the expected number of samples k that will fall in the region R will be:

$$E[k] = nP(\mathbf{x} \in R)$$

Now if the region R is small enough we can assume that $p(\mathbf{x})$ does not vary considerably within it, in which case:

$$P(\mathbf{x} \in R) = \int_R p(\mathbf{x}) d\mathbf{x} \simeq p(\mathbf{x}) V_R$$

where V_R is the volume of the region R over which we integrate. So we have:

$$\frac{E[k]}{n} = P(\mathbf{x} \in R) \simeq p(\mathbf{x}) V_R$$

thus we can get an estimate of the $p(\mathbf{x})$ probability density by:

$$p(\mathbf{x}) \simeq \frac{E[k]}{nV_R} \tag{1}$$

In kernel density estimation the estimation of $p(\mathbf{x})$ is done locally for each new example $\mathbf{x}$ on its neighborhood R using all the examples contained in the traning

set. If we assume that R is a hypercube with edge length h then its volume is given by $V_R = h^d$ where d is the dimensionality of our space, i.e. the number of attributes of our instances.
Now lets define a kernel function $K(\mathbf{x})$ as follows:

- $K(\mathbf{x}) = 1$ if $|x_j| \leq 1/2, j = 1, \ldots, d$
- $K(\mathbf{x}) = 0$ otherwise

$K(\mathbf{x})$ defines a *unit* hypercube centered at the origin of the axis, figure 1 shows such a hypecube. Then

$$K(\frac{\mathbf{x} - \mathbf{x}_i}{h})$$

is equal to one if $\mathbf{x}_i$ falls within the R hypercube of volume V_R centered at $\mathbf{x}$ and it is zero otherwise. So the expected number of training points $\mathbf{x}_i$ within the neighborhood R of $\mathbf{x}$ is given by:

$$E[k] = \sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x}_i}{h}) \tag{2}$$

Since the kernel given above is not smooth we will use for the exercise the gaussian kernel which, for the univariate and multivariate cases, is given by :

$$K_1(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x^2) \tag{3}$$

$$K_d(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} exp(-\frac{1}{2}\mathbf{x}^t\mathbf{x}) \tag{4}$$

To estimate the density function $p(\mathbf{x})$ we use equations 1,2 which give:

$$p(\mathbf{x}) \simeq \frac{1}{nV_R} \sum_{i}^{n} K(\frac{\mathbf{x} - \mathbf{x}_i}{h}) \tag{5}$$

where n the number of training points.
There are two ways to handle the case that $\mathbf{x}$ is a multivariate vector of dimension d.

- In the first one, we fit one kernel to each of the d dimensions, then equation 5 becomes:

$$p(\mathbf{x}) = \frac{1}{n \times V_R} \sum_{i}^{n} \prod_{j=1}^{j=d} K(\frac{x_j - x_{ij}}{h}) \tag{6}$$

- In the second approach we simply use the multivariate version of the kernel given by equation 4.

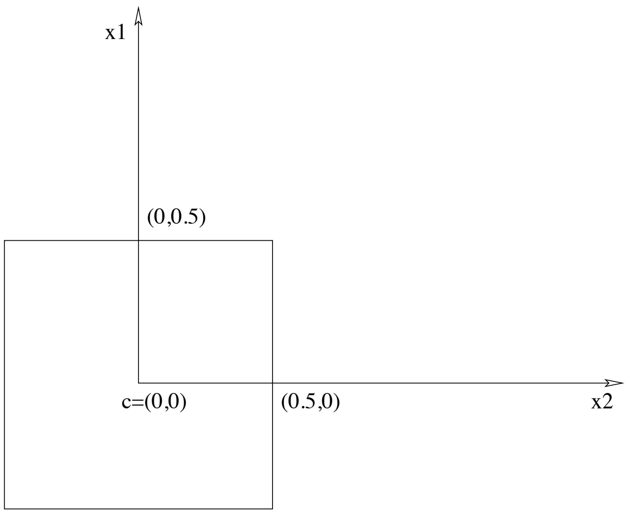


Figure 1: A hypercube centered at the origin of the axis, on a two dimensional space, with an edge length of one.

Let us examine for a moment the relation between the univariate normal distribution and the univariate kernel given by equation 5. When substituting in eq 5 the univariate kernel K_1 with its definition given in equation 3 we obtain:

$$p(x) = \frac{1}{n \times V_R} \sum_i \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}(\frac{x - x_i}{h})^2) \tag{7}$$

and since in the univariate case $V_R = h$ equation 7 becomes:

$$p(x) = \frac{1}{n} \sum_i \frac{1}{\sqrt{2\pi}h} exp(-\frac{1}{2}(\frac{x - x_i}{h})^2) \tag{8}$$

In other words $p(x)$ is the average of n univariate normal distributions each one centered in one of the training points x_i with a standard deviation of h.

The situation is the same in the multivariate case. Consider equation 5 where $K(\mathbf{x})$ is replaced by $K_d(\mathbf{x})$, $\hat{p}(\mathbf{x})$ is given by:

$$p(\mathbf{x}) = \frac{1}{n \times V_R} \sum_i \frac{1}{(2\pi)^{d/2}} exp(-\frac{1}{2} \frac{(\mathbf{x}-\mathbf{x}_i)^t}{h} \frac{(\mathbf{x}-\mathbf{x}_i)}{h}) \tag{9}$$

It is easy to show that

$$\frac{(\mathbf{x}-\mathbf{x}_i)^t}{h} \frac{(\mathbf{x}-\mathbf{x}_i)}{h} = (\mathbf{x}-\mathbf{x}_i)^t \mathbf{H}^{-1} (\mathbf{x}-\mathbf{x}_i) \tag{10}$$

where $\mathbf{H}$ is a diagonal matrix with all its diagonal elements equal to h^2, in which case the determinant $|\mathbf{H}| = h^{2d} = V_R^2$. Finally replacing V_R with $|\mathbf{H}|^{1/2}$, equation 9 now becomes:

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_i \frac{1}{(2\pi)^{d/2} |\mathbf{H}|^{1/2}} exp(-\frac{1}{2} (\mathbf{x}-\mathbf{x}_i)^t \mathbf{H}^{-1} (\mathbf{x}-\mathbf{x}_i)) \tag{11}$$

Which is just the average of n multivariated normal distributions, each one centered at a point x_i and all of them having the same covariance matrix $\mathbf{\Sigma} = \mathbf{H}$. A simple remark here, the determinant of the covariance matrix is a measure of the hypevolume of the data that gave rise to Σ.

Exercises

The final report should include a **detailed description of your observations,** e.g. comments on the forms of the density functions, the classification performance. Your code should be generic with detailed comments.

Introduction The TP is divided in three parts: The first part concerns the definition of the appropriate functions for probability density estimation using kernels and the study of the effect of the h parameter on a simple artificial set. The second concerns the application of the functions written previously on the iris dataset. The third is to apply the density estimation to a classification problem.

Exercise 1 Let your training set consist of the four following points :

$$\begin{array}{ccc} c_1 & : & 1,\ 1 \\ c_2 & : & 1,\ 4 \\ c_3 & : & 3,\ 2.5 \\ c_4 & : & 4,\ 2.5 \end{array}$$

Experiment with the following values of h : 0.3, 0.4, 0.5. You will find a file called kernel.py, it has some examples you might need, you can either complete it or make your own code from scratch. You have to create a regular set of points which cover the plane $[0, 5] \times [0, 5]$ and stores them in *testSet*. These

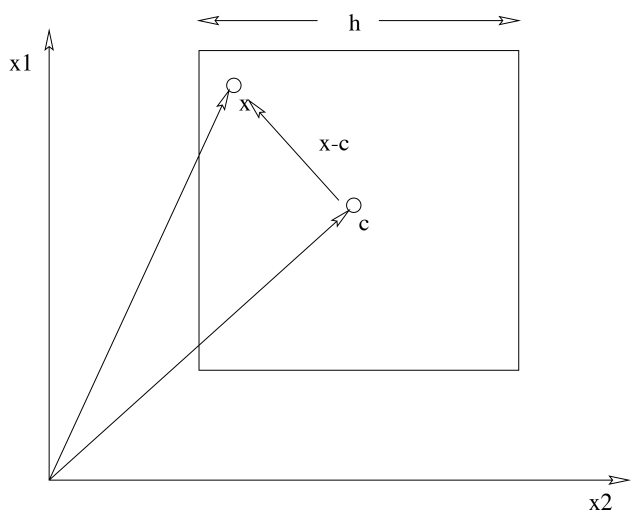


Figure 2: A hypercube centered at point c, on a two dimensional space.

points will be input into the density function so that we can compute its values in a regular way. The training set consists of the points given above and it is stored in variable c.

- Write a function that computes the gaussian kernel, with a center c_i and a size of hypercube h.
- Plot $p(x|c_i)$ for every $x \in testSet$ and every c_i (for h : 0.3, 0.4, 0.5) and comment your results.
- Plot $p(x)$ for every $x \in testSet$ (for h : 0.3, 0.4, 0.5) and comment your results.

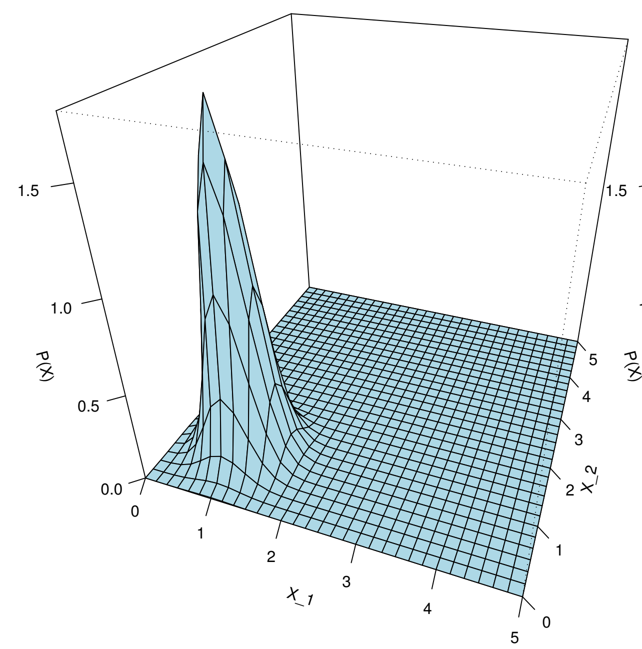
For h=0.3 you should get graphs similar to the graphs of the next pages. Discuss the effect of the size of h.

Exercise 2 Using the functions that you created above (exercise 1) work with the iris dataset and:

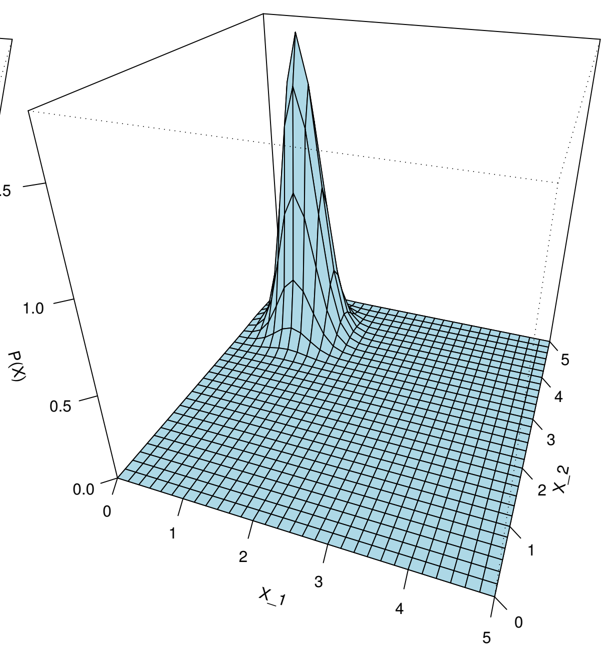
- Plot the class conditional density of each attribute (figure 3)
- For a given pair of attributes draw the two dimensional density for each class (e.g. figures4,5,6).
- Experiment with at least three different values of the h parameter and comment on your findings.

Exercise 3 (TP1-Naive Bayes Exercise 2c) Implement the Naive Bayes on iris dataset but now instead of assuming normal distribution estimate the probability distribution from the data (using kernel density estimation, h : 0.3, 0.4, 0.5) Discuss (in details) the effect of the h parameter on the classification accuracy of the algorithm and compare with the results that you had in tp1 (NB).

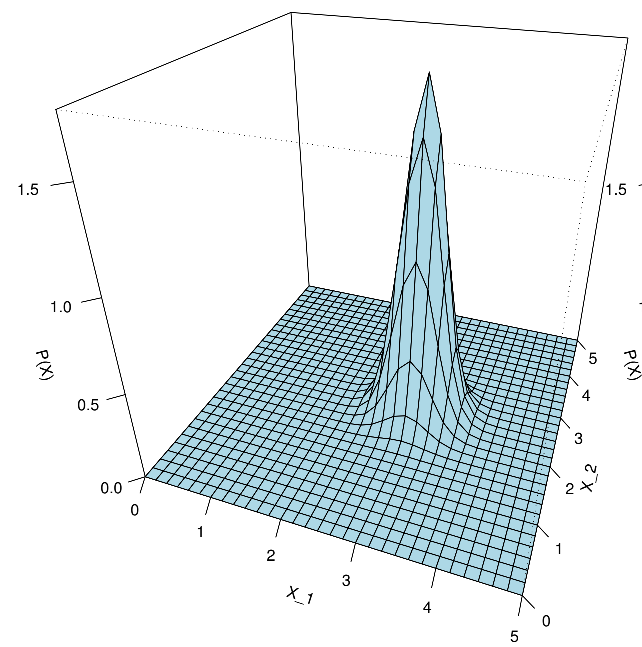
Kernel Function at point : (1,1). h: 0.3



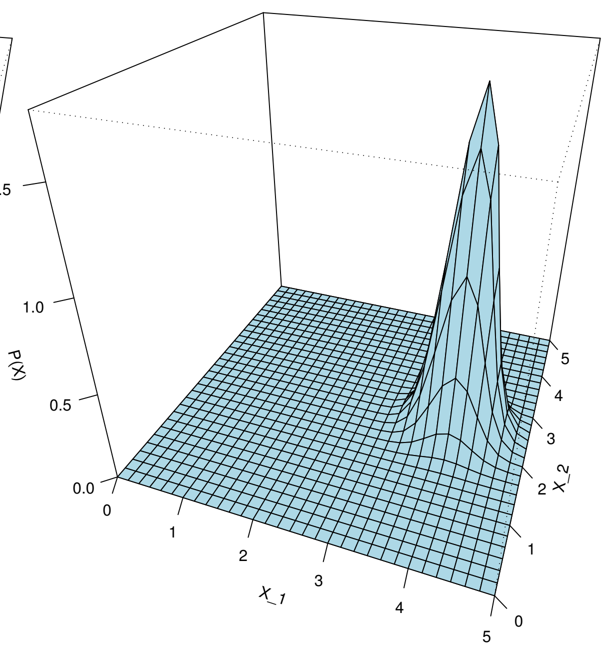
Kernel Function at point : (1,4). h: 0.3



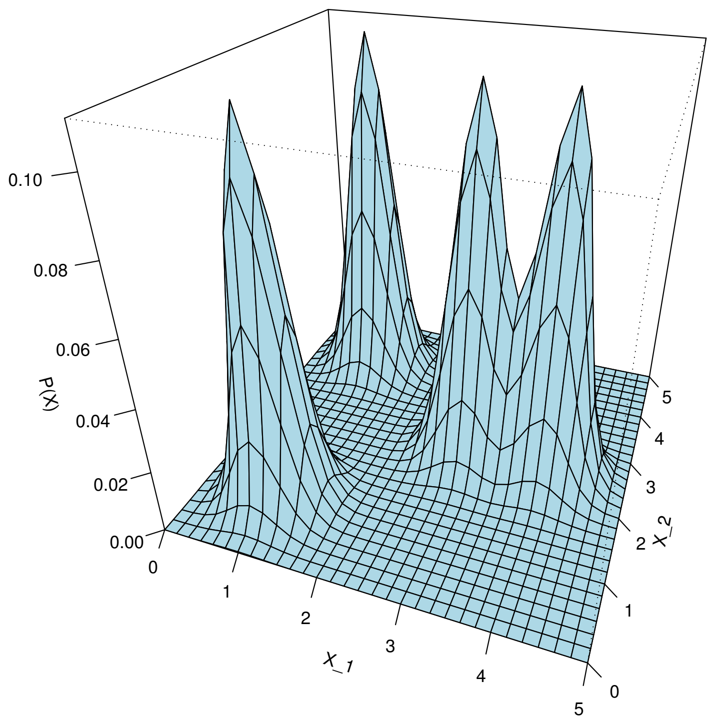
Kernel Function at point : (3,2.5). h: 0.3



Kernel Function at point : (4,2.5). h: 0.3



Density Function p(x), h : 0.3



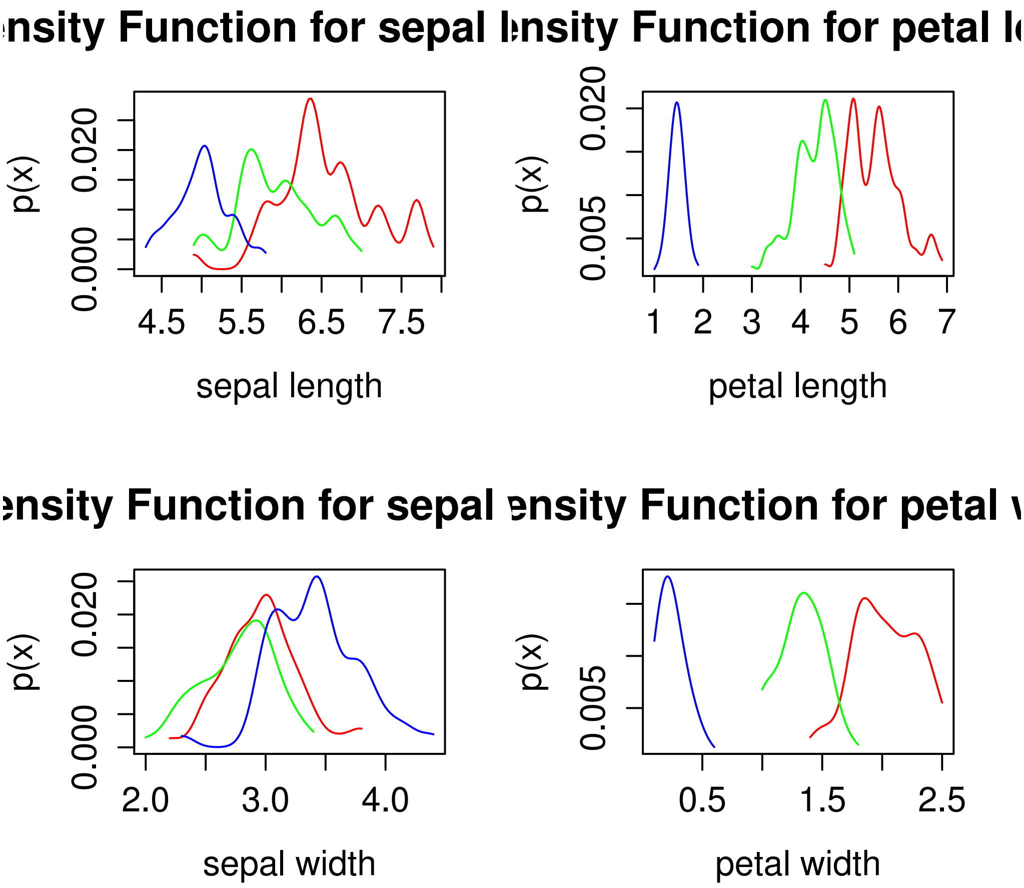


Figure 3: The class conditional densities of each attribute of the iris datasets. The edge h of the hypercube was set to 0.1

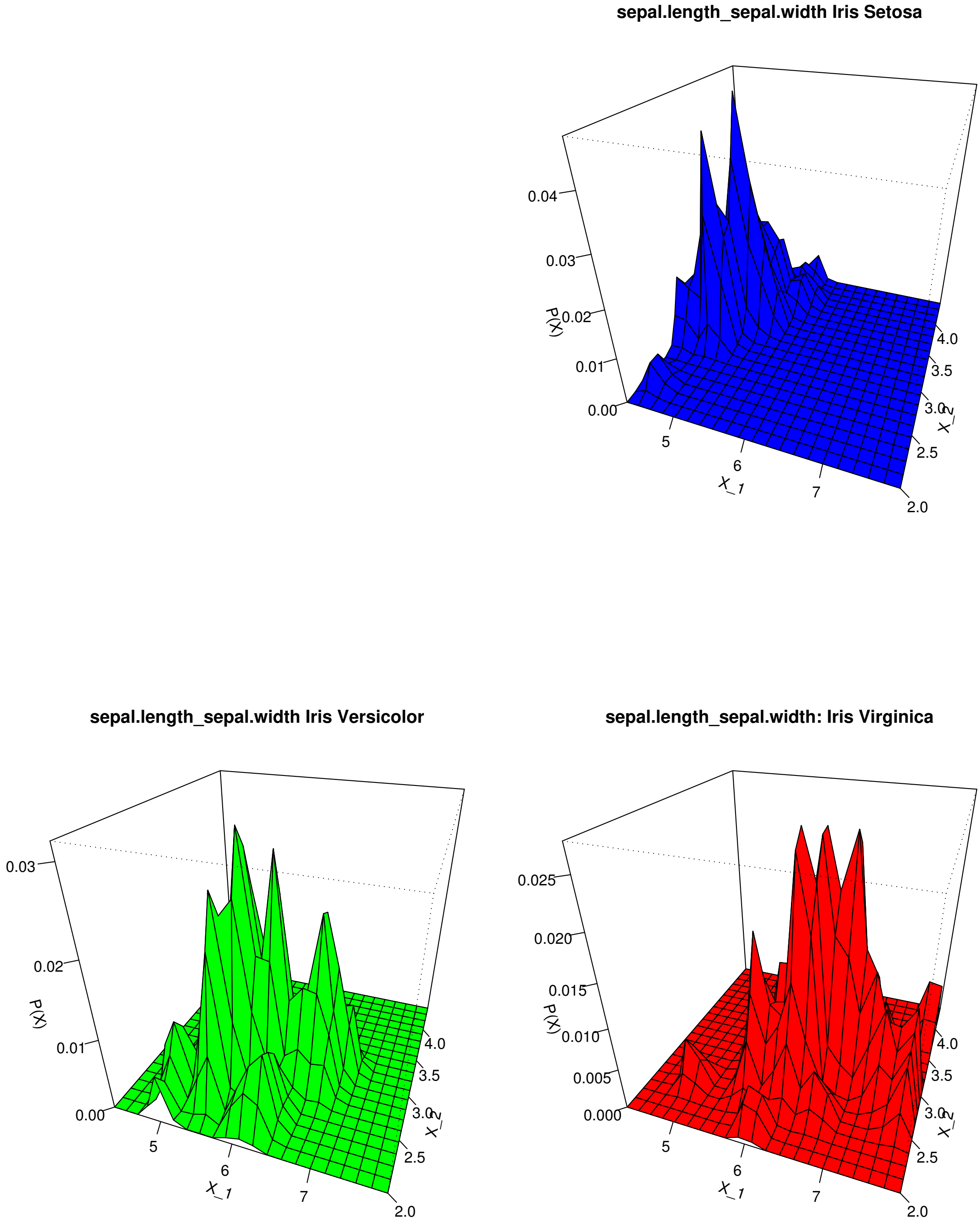


Figure 4: The two dimensional class conditional densities for the pair of attributes: (Sepal Length, Sepal Width)

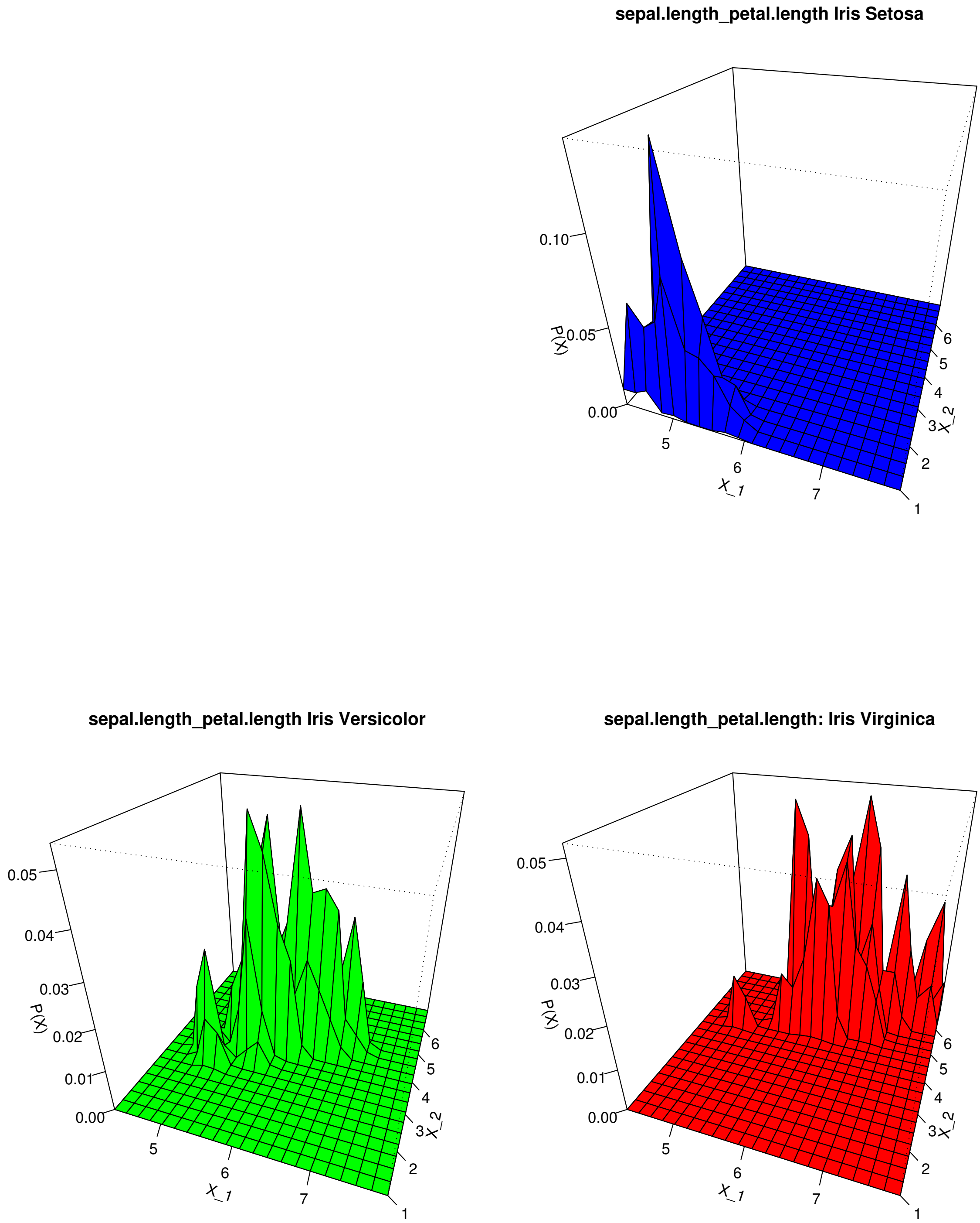


Figure 5: The two dimensional class conditional densities for the pair of attributes: (Sepal Length, Petal Length)

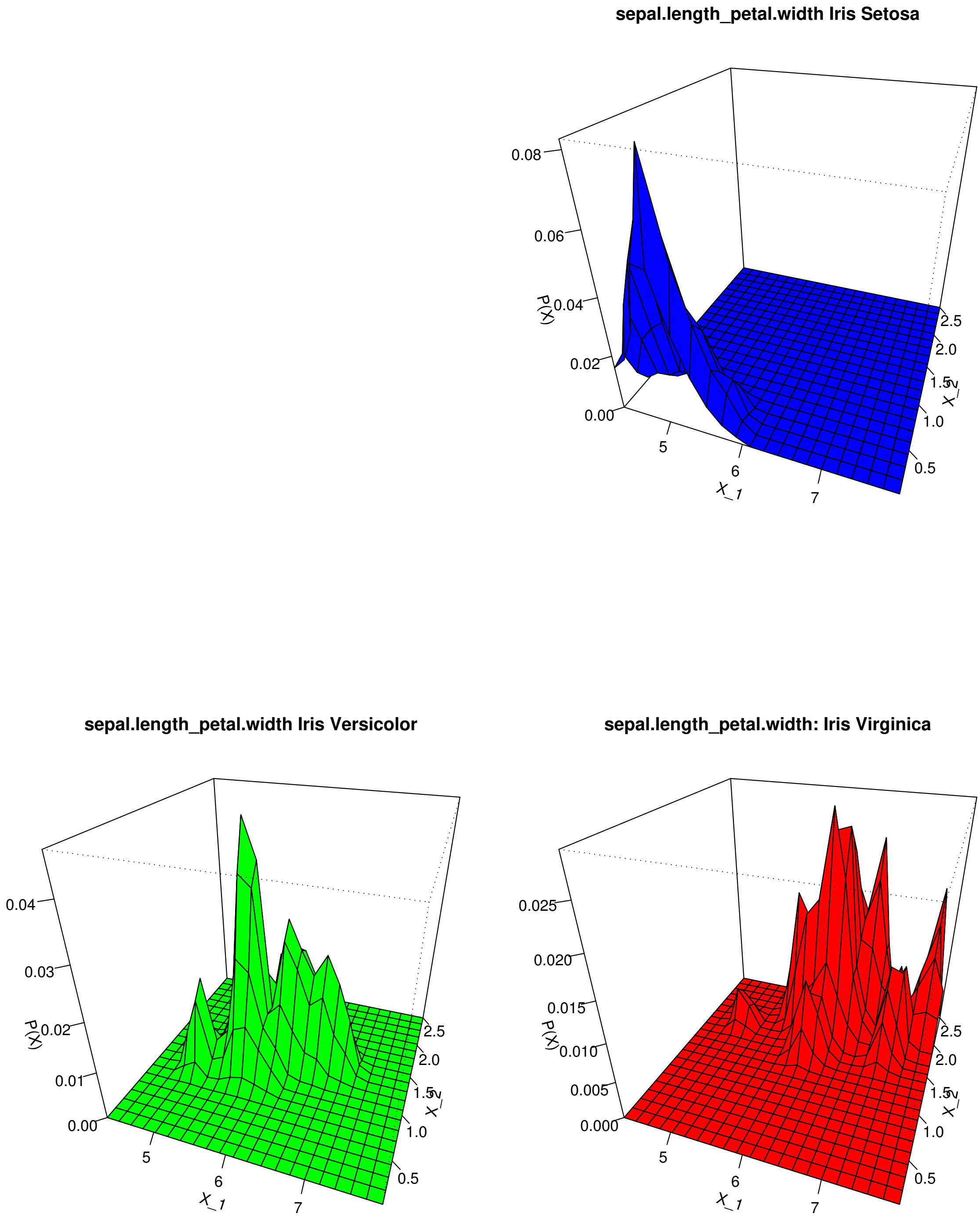


Figure 6: The two dimensional class conditional densities for the pair of attributes: (Sepal Length, Petal Width)